REMARKS

I. STATUS OF THE CLAIMS

A minor amendment is made herein to claim 20, to correct a typographical error.

Claims 5 and 26 are allowed.

Claim 23 is "objected to".

Claims 24 and 25 are canceled herein.

In view of the above, it is respectfully submitted that claims 5 and 11-23 and 26 are currently pending.

II. REJECTION OF CLAIMS 11-22, 24 AND 25 UNDER 35 USC 103 AS BEING UNPATENTABLE OVER YAGASAKI (US PATENT NO. 5,428,396) IN VIEW OF KOVALEV (US PATENT NO. 6,339,616)

Claim 11 recites decoding means for decoding the motion vector of the target block by using a result of the prediction made by said predicting means with a decoding method *selected from at least two different decoding methods* based on a result of a determination made by said determining means, *wherein said at least two different decoding methods use different codes, respectively, to decode the same motion vector.*

Yagasaki does not disclose or suggest these features.

Yagasaki is directed to reducing the size of a reference table. See, for example, column 19, lines 8-9, of Yagasaki. To achieve this objective, Yagasaki uses a "fold-back" method as shown in Table 5 of Yagasaki and described in column 19, line 3, through column 21, line 3, of Yagasaki.

For example, in Table 5 of Yagasaki, the code "0000 0011 011" corresponds to the value "-15". However, to reduce the size of the reference table, the code "0000 0011 011" in Table 5 also corresponds to the value "17". Therefore, Table 5 uses the same code to correspond to two different values. Yagasaki uses the motion vector value for the preceding block to determine which value, for example, "-15" or "17", is the proper value. See, for example, column 19, line 64, through column 20, line 6, of Yagasaki.

In order to make it easy to understand the encoder and decoder procedure of "fold-back" method, more detailed explanation is described below in case of Table 5.

Here, the following values are defined.

- MVx is a x-component of a motion vector. The range is limited to [-16:15] inclusive.
- MVPx is the motion vector value for the proceeding block of MVx. The range is limited [-16:15] inclusive.
- MVDx is a differential value of MVx by subtracting MVPx from MVx, that is,

MVDx = MVx – MVPx

The range of MVDx is [-31: 31]. That is, MVDx is equal to -31 when MVx is -16 and MVPx is 15, and MVDx is equal to 31 when MVx is 15 and MVPx is -16.

On the encoder, MVx and MVPx are available. MVDx is calculated by subtracting MVPx from MVx. Then MVDx is encoded using VLC code defined Table 5. Because the range of MVDx is [-31:31], that is, the kinds of MVDx values is 63 and the entry number of Table 5 is 32, 31 (=63-32) kinds of MVDx values should share the same VLC code as described in Table 5.

On the decoder, MVPx can be calculated using the previously received motion vectors. Using MVPx and received MVDx from the encoder, MVx is reproduced in the following way.

1) When a VLC code is received, the SMALL value and LARGE value of the corresponding VLC code are extracted from Table 5.

2) First of all, the summation of MVPx (i.e., the motion vector value for the proceeding block of MVx) and SMALL is calculated. If the summation is in the range of [-16:15], which is the range of MVx, then this summation is assigned to MVx.

3) Otherwise(if the summation is not in the range of [-16:15]), the summation of MVPx and LARGE is assigned to MVx.

In this process, it is guaranteed that only the one of (MVPx+SMALL) and (MVPx + LARGE) is in the range of [-16:15].

Here, please consider the following two cases.

<u>Case 1</u>

<on the encoder>

- MVx is 1 and MVPx is -16.
- In this case, MVDx is equal to 17 (MVx – MVPx).
- From Table 5, the VLC code for 17 is “0000 0011 011”

<on the decoder>

- VLC code “0000 0011 011” is received.
- MVPx is -16 (the same as the encoder)
- SMALL is -15 and LARGE is 17.
- MVPx + SMALL = (-16) + (-15) = -31. This is not in the range of MVx, that is [-16:15].
- MVPx + LARGE = (-16)+17 = 1. This is in the range of [-16:15]. Then the value “1” is assigned to MVx. This value is the same value of the encoder.

Case 2

<on the encoder>

- MVx is -16 and MVPx is -1.
- In this case, MVDx is equal to -15 (MVx – MVPx).
- From Table 5, the VLC code for 17 is "0000 0011 011"

<on the decoder>

- VLC code "0000 0011 011" is received.(The same VLC code as case 1)
- MVPx is -1 (the same as the encoder)
- SMALL is -15 and LARGE is 17.
- MVPx + SMALL = (-1) + (-15) = -16. This is in the range of MVx, that is [-16:15]. Then the value "-16" is assigned to MVx. This is the same value of the encoder. (In this case, MVPx + LARGE = (-1)+17 = 16, and it is not in the range of [-16:15])

As described above, the "fold-back" method can reproduce the encoded motion vector correctly on the decoder side using the shrunk table(such as Table 5) by limiting the range of reproduced motion vector.

Therefore, Yagasaki always uses the same code for decoding the same value. For example, the code "0000 1010" is always used for decoding the value "5". As can be seen from this example, Yagasaki discloses that only one decoding method is used. Yagasaki does not select a decoding method from at least two different decoding methods that use different codes, respectively, to decode the same motion vector..

As an example, with various embodiments of the present invention, based on a result of a determination made by a determining means, a decoding method might be selected which, for example, uses a specific code corresponding to the value "5". However, based on a different result of a determination made by a determining means, a different decoding method might be selected which, for example, uses a different specific code corresponding to the value "5". For example, in FIG. 11 of the present application, "variable length codes 1" uses the code "0000010010" corresponding to the value "5", whereas "variable length codes 2" uses the code ""101000" corresponding to the value "5".

This operation is significantly different than Yagasaki, where the same code (e.g., "0000 1010" in Table 5 of Yagasaki) is always used for the value "5". Please note that the claimed invention is not intended to be limited in any way to the examples described herein.

Therefore, Yagasaki does not disclose or suggest the present invention as recited, for

example, in claim 11.

Moreover, please note that claim 11 includes specific recitations relating to degrees of non-uniformity of motion vectors. Yagasaki does not include any disclosure relating to degrees of non-uniformity of motion vectors.

In the paragraph that spans pages 2-3 of the outstanding Office Action, the Examiner appears to indicate that the assignment of the same code to two different values in Table 5 of Yagasaki, corresponds to non-uniformity of motion vectors. However, as can be seen from the above-described example in Yagasaki, the assignment of the same code to two different values has nothing to do with non-uniformity of motion vectors. Non-uniformity of motion vectors can be understood, for example, from the examples in FIGS. 4, 5 and 13, and the examples in the disclosure on page 31, line 11, through page 33, line 20, of the specification of the present application.

Moreover, in the paragraph that spans pages 2-3 of the outstanding Office Action, the Examiner refers to column 18, lines 1-13, and column 19, lines 11-63, of Yagasaki. It appears that the Examiner believes this portion of Yagasaki relates to degrees of non-uniformity of motion vectors. From a review of this portion of Yagasaki, it appears that the Examiner would correlate the values LARGE and SMALL in the above-described examples of Yagasaki to degrees of non-uniformity of motion vectors. However, as can be seen from the above-described examples of Yagasaki, the values LARGE and SMALL do not indicate any type of degrees of non-uniformity of motion vectors, and are instead used for a completely different purpose.

In the outstanding Office Action, the Examiner concedes that Yagasaki does not specifically disclose the use of two different decoding methods that use different codes to decode the same motion vector.

Instead, the Examiner asserts that Kovalev teaches the use of two different decoding methods that use different codes to decode the same motion vector. The Examiner then combines Yagasaki with Kovalev to reject the claimed invention.

In Kovalev, pixels are classified as either static, new or directionally estimated. The classifications of static, new and directionally estimated are described, for example, in column 5, lines 40-65, of Kovalev. Table 1 in column 8 of Kovalev shows values [1-8] corresponding to directions for directionally estimated pixels, value 0 for static pixels, and value -1 for new pixels. See, for example, column 8, lines 39-42, of Kovalev.

FIG. 7 of Kovalev discloses a pixel classification unit 706 for classifying pixels. Then, depending on the classification, the pixels are encoded by an run-length (RL) encoder 708 for static pixel (column 8, lines 11-12), a VLC encoder 710 for directionally estimated pixel (column 7, lines 1-3), or a Delta Value (New Pixel) encoder 712. See column 17, lines 13-43, of Kovalev. Column 17, lines 30-33, clearly indicates the specific encoder that is used for a pixel is

dependent on the classification of the pixel as either static, new or directionally estimated.

Therefore, the encoding method in FIG. 7 of Kovalev is based on the classification of the pixel as being either status, new or directionally estimated.

In the Office Action, the Examiner refers to elements in FIG. 8 of Kovalev. In FIG. 8 of Kovalev, a demultiplexing unit 802 is coupled to a decoding unit 804. Decoding unit 804 includes an RL decoder 806, a VLC decoder 808 and a Delta Value decoder 810. See also column 18, lines 11-21, of Kovalev.

The arrangement of the demultiplexing unit 802 and decoders 806, 808 and 810 in FIG. 8 of Kovalev is opposite that of pixel classification unit 706 and encoders 708, 710 and 712 in FIG. 7. Therefore, it should be understood that whether a pixel is decoded by RL decoder 806, VLC decoder 808 or Delta Value decoder 810 is dependent on the classification of the pixel. That is, whether a pixel is decoded by RL decoder 806, VLC decoder 808 or Delta Value decoder 810 *is dependent on whether the pixel is classified as static, new or directionally estimated.*

Therefore, it appears that the value of a pixel (as indicated in Table 1 of Kovalev) is sent to the decoding side so that the proper one of RL decoder 806, VLC decoder 808 and Delta Value decoder 810 is used to decode a respective pixel, *in accordance with the classification of the pixel as being static, new or directionally estimated.*

Therefore, Kovalev does not disclose or suggest selecting a decoding method based on a result of a determination made by a determining means that determines accuracy of a prediction made by a predicting means based on degrees of non-uniformity of motion vectors of a plurality of blocks adjacent to the target block as recited, for example, in claim 11. Instead, as indicated above, Kovalev selects a decoder in accordance on a classification of a pixel as static, new or directionally estimated.

Accordingly, even if Kovalev was combined with Yagasaki, the combination would not disclose or suggest the present invention as recited, for example, in claim 11.

Further, claim 11 specifically recites decoding *the motion vector of the target block.* The decoding performed in FIG. 8 of Kovalev is not of a motion vector of a target block. For example, FIG. 7 and the disclosure in column 17, lines 1-9, of Kovalev, indicate that the data being encoded is data from an image data source, such as a video camera, a still image digital camera, computer animation, a storage medium, a video conferencing link, etc. This encoded data is then decoded in FIG. 8. No disclosure in Kovalev indicates that a motion vector of a target block is being decoded.

In summary, the manner of which a decoder is selected in Kovalev is completely different than that recited, for example, in claim 11. In addition, the data that is being decoded in Kovalev is different than that decoded, for example, in claim 11.

Moreover, as Kovalev is directed to a specific type of encoding/decoding in which pixels

are classified as static, new or directionally estimated, and there is no disclosure in Kovalev that relates to encoding/decoding of a motion vector of a target block, it is respectfully submitted that it is improper for the Examiner to modify the encoding/decoding of Kovalev for use with motion vectors of a target block.

Due to at least the above-described deficiencies in Kovalev, it is respectfully submitted that Kovalev and Yagasaki should not be combined in the manner proposed by the Examiner.

The above comments are specifically directed to claim 11. However, it is respectfully submitted that the comments would be helpful in understanding various differences of various other claims over the cited references.

In view of the above, it is respectfully submitted that the rejection is overcome.

III. UNACKNOWLEDGED REFERENCE IN IDS

The Office Action mailed March 14, 2007, included an acknowledged Form PTO-1449 of the IDS filed November 28, 2006. However, the Examiner did not "initial" reference AM on the Form PTO-1449.

Therefore, during an Examiner Interview conducted at the USPTO on April 12, 2007, the Examiner "initialed" reference AM and provided a copy of the initialed Form PTO-1449 to the applicant's representative.

IV. CONCLUSION

In view of the above, it is respectfully submitted that the application is in condition for allowance, and a Notice of Allowance is earnestly solicited.

If any further fees are required in connection with the filing of this response, please charge such fees to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: April 19, 2007

By: [signature]
Paul I. Kravetz
Registration No. 35,230

1201 New York Avenue, NW, 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501